

Software Metrics - The Key to Quality Software on the NCC Project

Patricia J. Burns
Computer Sciences Corporation
10110 Aerospace Road Seabrook, Maryland
301.794.1640 (o) 301.552.3272 (fax)

SUMMARY

Network Control Center (NCC) Project metrics are captured during the implementation and testing phases of the NCCDS software development lifecycle. The metrics data collection and reporting function has interfaces with all elements of the NCC project. Close collaboration with all project elements has resulted in the development of a defined and repeatable set of metrics processes. The resulting data are used to plan and monitor release activities on a weekly basis. The use of graphical outputs facilitates the interpretation of progress and status.

The successful application of metrics throughout the NCC project has been instrumental in the delivery of quality software. The use of metrics on the NCC Project supports the needs of the technical and managerial staff. This paper describes the project, the functions supported by metrics, the data that are collected and reported, how the data are used, and the improvements in the quality of deliverable software since the metrics processes and products have been in use.

NCC PROJECT OVERVIEW

The NCC is an element of the National Aeronautics and Space Administration (NASA) Spaceflight Tracking and Data Network (STDN). The STDN is a worldwide complex designed to provide tracking and data acquisition support to manned and unmanned spacecraft in low-earth orbit. It is composed of the Space Network (SN) and the Ground Network (GN). The STDN has evolved into a network that currently uses the Tracking and Data Relay Satellite System (TDRSS) as the primary source of support for orbiting spacecraft. The NCC Project Office, Goddard Spaceflight Center (GSFC) Code 530, is responsible for the support and maintenance of the current NCC Data System (NCCDS). The NCCDS performs network scheduling, acquisition and tracking support, data quality assurance, performance monitoring, and overall coordination of the STDN.

The NCCDS is maintained by the NCC Project within the Computer Sciences Corporation (CSC) System Sciences Division (SSD) Networks Technology Group (NTG). The maintenance and enhancement of the NCCDS is performed as part of the System

PRECEDING PAGE BLANK NOT FILMED

Engineering and Analysis Support (SEAS) contract. These responsibilities have been distributed among several tasks: Software Development, Software Maintenance, Integration Test, System Test, System Engineering, Metrics, Configuration Management, and Product Assurance.

NCC SOFTWARE METRICS DEFINED

On the NCC Project, the metrics program consists of defined, documented processes and measurement tools that provide a quantitative and qualitative representation of the status of a software build or release. Data items including measures of quantity, scheduled and actual progress, number of iterations, and defect information are collected, stored, and reported weekly to provide a snapshot of progress and work yet to be accomplished. All of the tools described in this paper are implemented using a spreadsheet on a 386 or 486 compatible personal computer. The effort expended on project metrics task activities has varied. At peak staffing of the NCC project, 3 full-time analysts were assigned to the metrics activity.

METRICS IN NCCDS SOFTWARE LIFECYCLE

The NCC metrics task originated coincident with the software development activity at the outset of the NCCDS upgrade to support the Second TDRSS Ground Terminal (STGT). It remains a responsibility of the Software Development Department. While the bulk of the data collected and reported by metrics is related to software development, the integration test, system test, and software maintenance tasks are also primary metrics customers. Other project tasks utilize metrics data as an ancillary function.

The phases of the NCCDS software development lifecycle are illustrated in Figure 1. The major phases of the lifecycle are requirements identification, design and implementation, and testing. The testing phase is subdivided into integration, system and acceptance testing activities. Each of these phases of testing is performed by a different test group. Milestones that hallmark the software development lifecycle are: system requirements review (SRR), preliminary design review (PDR), critical design review (CDR), integration test readiness review (ITRR), system test readiness review (STRR), acceptance test readiness review (ATRR), and the operations readiness review (ORR). The metrics group provides active support during the design and implementation phase, the integration test phase, and the system test phase.

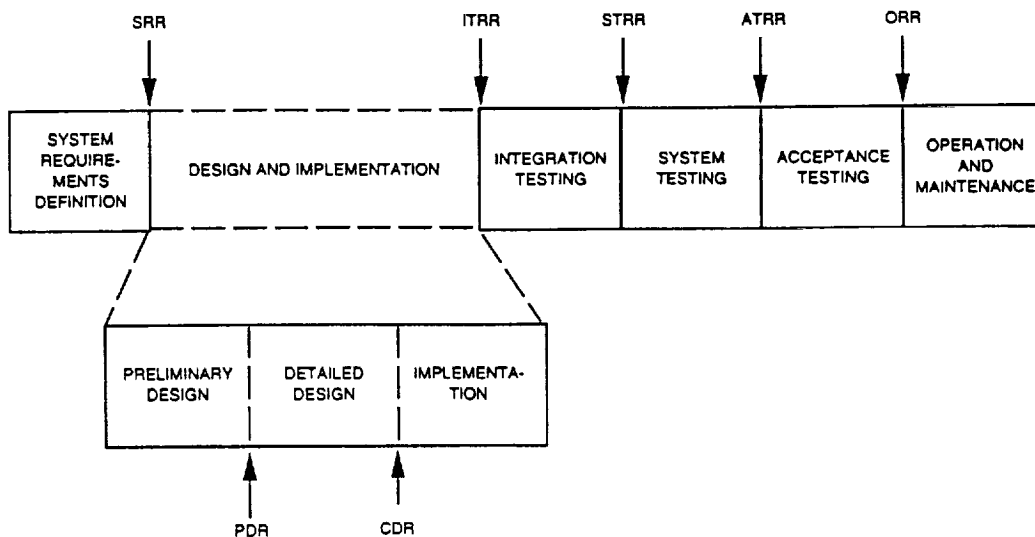


Figure 1. NCCDS Software Development Lifecycle

The lifecycle for NCCDS software maintenance, illustrated in Figure 2, differs slightly from the above. The major phases can be defined as contents definition, problem resolution and implementation, and testing. The initial phase in the maintenance lifecycle is defined as contents definition as opposed to requirements identification as contents are generally not major system enhancements. They are predominantly fixes to anomalies reported during operational use that have been scheduled for a particular maintenance release. Milestones that hallmark the maintenance lifecycle are an official contents letter, followed by the test readiness reviews listed above. Formal reviews such as SRR, PDR, and CDR are not included in the software maintenance lifecycle.

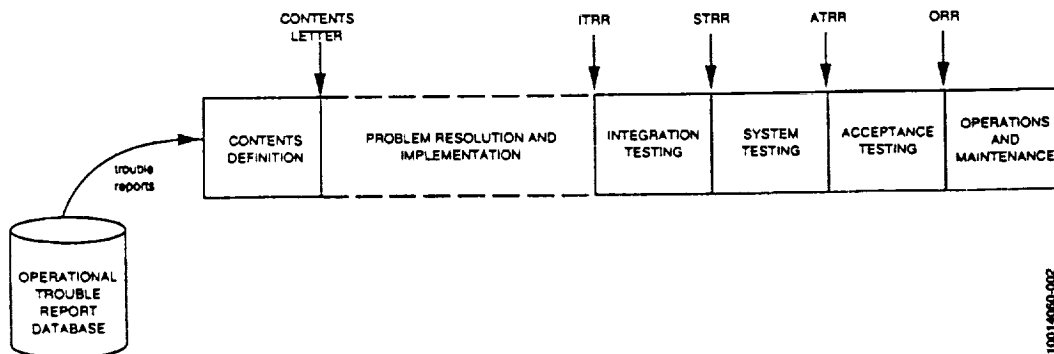


Figure 2. NCCDS Software Maintenance Lifecycle

METRICS DATA COLLECTION AND REPORTING

When using metrics to support software implementation and testing, the objective is to establish a baseline plan for the work to be done, and then perform work according to the plan. Generally, this approach is the same for development, maintenance and testing with slight variations according to the nature of the task being supported. Data flow between development and maintenance customers and the metrics group is illustrated in Figure 3.

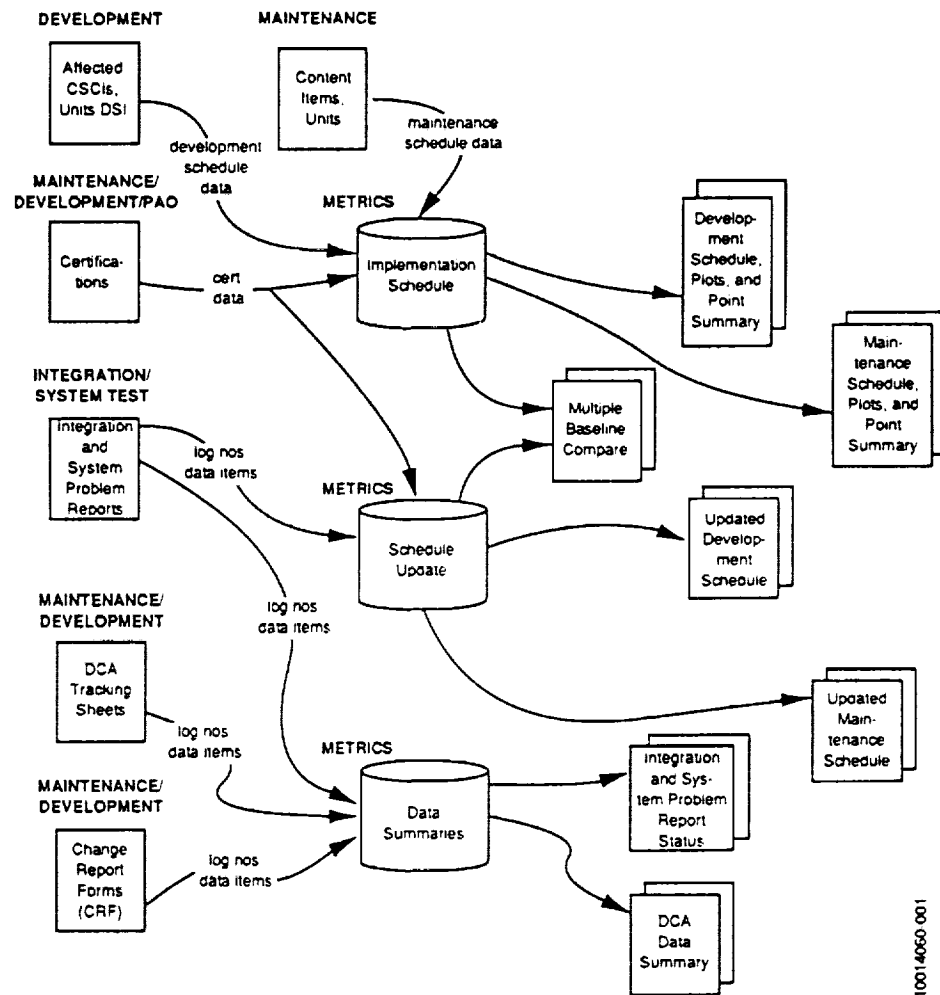


Figure 3. Data Flow - Development, Maintenance and Metrics

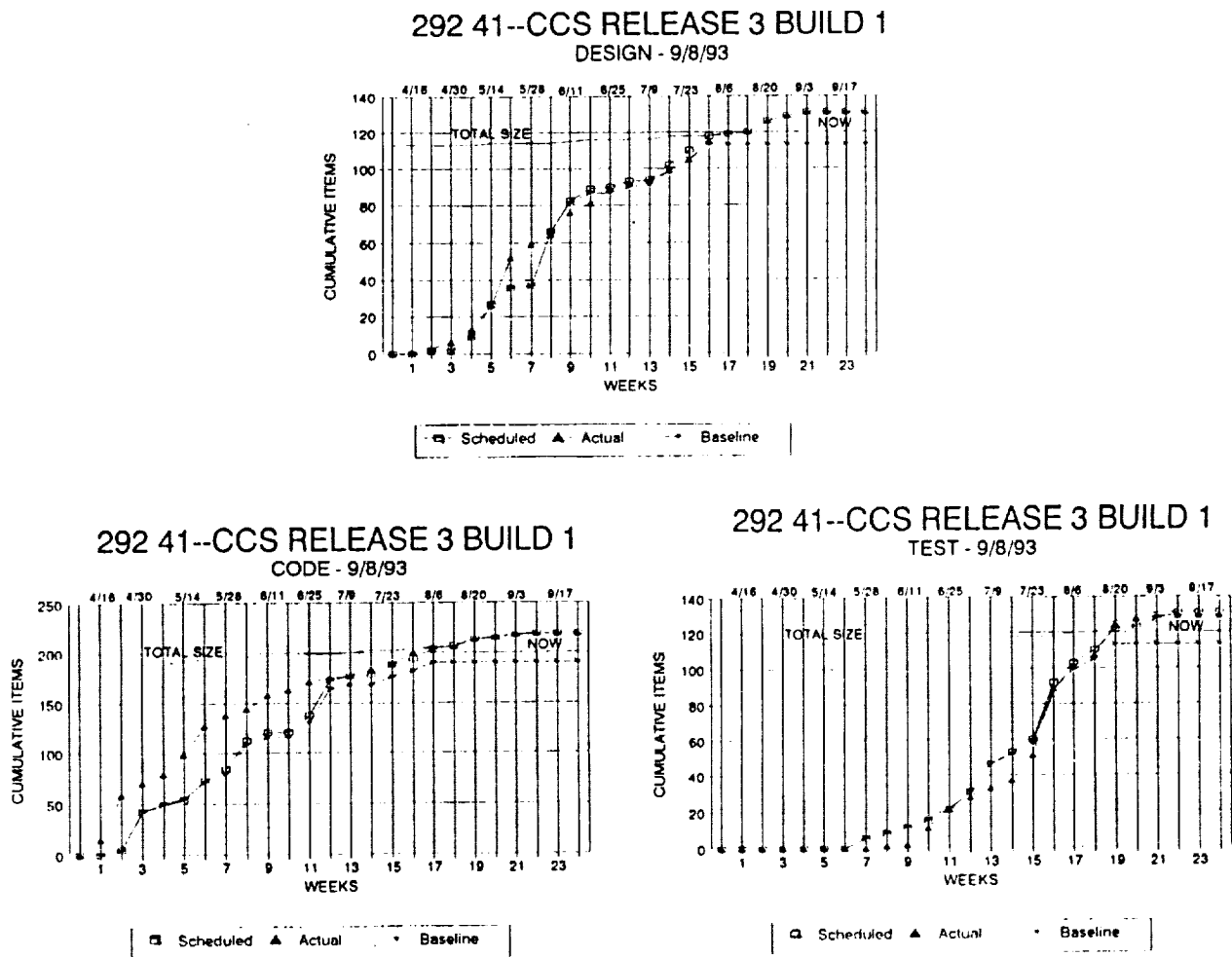
Development and Maintenance Support: Implementation Phase

The primary products produced by Metrics during a development or maintenance software implementation are schedules, plots, and points summaries. To begin the process of metrics support of a build or release, both the development and the maintenance organizations must submit lists of the items (units, displays, etc) that are planned for change in the build or release along with an estimate of the size of each item in delivered source instructions (DSI). Development items are identified according to the computer software configuration item and software component (CSCI and SC) along with the item name, and the anticipated impact of the change (a percentage of the DSI.) Maintenance items are identified according to the problem report number, or some other form identifier, along with the item name. This information is used to build an *Implementation Schedule* spreadsheet. A sample portion of a development implementation schedule is shown in Figure 5 below.

of of		06/05/93		A		B		C		D		TEST	
LA Number		A		B		C		D		TEST		TEST	
CSCI	SC#	MODULE NAME	CHGDATE	TYPE	IMP	DSI	DESIGN	TEST PLAN	DESIGN	CODE	TEST	TEST	TEST
							Plan	Actual	Plan	Actual	RESP P	RESP I	RECEIPT
5302	0000	CSTWAG	04/12/93	U	A	80	06/30/93	06/15/93	06/30/93	06/15/93	D E	M SP	06/30/93
5302	0000	CSTACET	04/12/93	P	C	34					D E	M SP	06/30/93
5302	0000	CSTSC	04/12/93	U	C	94	06/30/93	06/15/93	06/30/93	06/15/93	D E	M SP	06/30/93
5302	0000	CSTSERVP	04/12/93	U	A	132	06/30/93	06/15/93	06/30/93	06/15/93	D E	M SP	06/30/93
5302	0000	CSTBLFND	04/12/93	U	C	88	06/30/93	06/15/93	06/30/93	06/15/93	D E	M SP	06/30/93
5302	0000	CSTTDNS	04/12/93	U	A	88	06/30/93	06/15/93	06/30/93	06/15/93	D E	M SP	06/30/93
5302	0000	CDUPM	04/12/93	P	C	17					D E	M SP	06/30/93
5302	0000	CSTVSP	04/12/93	P	A	82	06/30/93	06/15/93	06/30/93	06/15/93	D E	M SP	06/30/93
5302	0000	CTDRSC	04/12/93	P	A	30					D E	M SP	06/30/93
5302	0000	CTDRSM	04/12/93	P	A	12					D E	M SP	06/30/93
5302	0000	CTDRTAB	04/12/93	U	H	43	06/30/93	06/15/93	06/30/93	06/15/93	D E	M SP	06/30/93
5302	0000	CYSPCM	04/12/93	P	C	16					D E	M SP	06/30/93
5302	0000	CYSPCT	04/12/93	P	C	12					D E	M SP	06/30/93
5304	0000	GBLDEC	04/12/93	P	C	8					A M	D R	
END		DB TOTAL		17844	131	131	131	131	131	218	218	131	138

Figure 5. Development Implementation Schedule

Included are estimated dates for the design, code and test certification of each item on the schedule. The certification process is defined by the SEAS Software System Development Methodology (SSDM) to record completion of the design, code and test of items in a build or release. Weekly, as the actual certifications are completed, the completion dates are added to the implementation schedule. *Design, Code and Test Plots* are generated that contrast the planned certification progress against the actual certification progress. Each of these graphs also shows growth in the total number of items to be certified. Sample design, code and test plots are shown in Figure 6.



Lastly, a tally of *Points*, Figure 7, is calculated based on the total items to complete and the total completed. Points indicate the amount of credit that can be taken towards the completion of the activity. A different number of points are assigned to each activity - design, code, and test certification - by management. As the size of a build or release grows so does the number of total points. This is reflected in the difference between baseline size and current size, and the points added. Points are calculated weekly and included in the implementation schedule data summary. Each of these products, the schedule, plots, and points summary supports the timely delivery of quality software by providing detailed and high-level feedback on progress and remaining work.

SUMMARY DATA									
as of 07/23/93									
3329241									
Original Start Date:		04/16/93							
Original End Date:		09/24/93							
Projected End Date:		09/24/93							
Number of weeks elapsed		15.0							
Weeks to Original End:		9.0							
Weeks to Projected End:		9.0							
	Estimated Size		Percent Growth	Completed		Yet To Complete		Planned To Be Complete	Percent Ahead of Plan
	Baseline	Current		Number	Percent	Number	Percent		
DESIGN	113	118	4 %	100	85 %	18	15 %	118	-15
TEST PLAN	113	118	4 %	89	75 %	29	25 %	118	-25
CODE	190	202	6 %	184	91 %	18	9 %	194	-5
TEST	113	118	4 %	48	41 %	70	59 %	92	-48
Module Test	0	0	NA %	0	NA %	0	NA %	0	NA
spare 3	0	0	NA %	0	NA %	0	NA %	0	NA
spare 2	0	0	NA %	0	NA %	0	NA %	0	NA
spare 1	0	0	NA %	0	NA %	0	NA %	0	NA
Total Points	1438	1516	5 %	1232	81 %	284	19 %	1432	-14
Activity	Points	FISCAL MONTH POINT SUMMARY			CUMULATIVE		POINTS		
		Month	Planned	Completed	Planned	Completed	%	Added	
DESIGN	4	APR	176	304	176	304	73%	0	
TEST PLAN	0	MAY	314	484	490	788	61%	30	
CODE	4	JUN	636	336	1126	1124	-0%	24	
TEST	2	JUL	306	108	1432	1232	-14%	24	
Module Test	0	AUG	84	0	1516	0	0%	0	
spare 3	0	SEP	0	0	0	0	0%	0	
spare 2	0								
spare 1	0	TOTAL	1516	1232				78	
Total	10								

Figure 7. Development Points Summary

Development and Maintenance Support: Testing Phase

At the completion of an implementation phase, the build or release baseline is delivered to Configuration Management (CM) through the Product Assurance Office (PAO) to be placed under configuration control. Once a build or release has been placed under CM control, changes made to the baseline are tracked through the problem reporting system defined in the NCC Standards and Procedures.

The implementation schedule used to account for all items planned for the build or release is now revised to become the *Schedule Update*. The schedule update is used to track the impact of problem report changes on specific items in a build or release as it progresses through testing. The impacting problem report, recertification information, new DSI counts, and variance in the weighted amount of change are recorded on the schedule update. By referencing the schedule update, management and technical staff can identify which software components are being impacted by problem reports and determine if additional build or release items require attention. An enhancement for cross-checking between different builds and releases is the recently developed data summary, the *Multiple Baseline Compare*. The compare shows which software items are being updated simultaneously for different builds or releases. This information is critical in the NCCDS development and maintenance environment to insure the integrity of software products. Implementation activities for different builds and releases sometimes proceed in parallel, requiring that each baseline be updated separately, but yet remain consistent with each other. For example, although a development build may be initiated before a maintenance release, the full development release may not be delivered to the customer until after delivery of the maintenance release. All changes made in the maintenance release must also be present in the superseding development release. The compare report makes the development and maintenance programming staff aware of the need to merge changes from one implementation effort into others as applicable, thus preventing the loss of upgrades and fixes from one baseline to another.

Integration and System Test Metrics Support

There are two levels of testing performed by NCC project staff against each software build or release - integration testing and system testing. The data flow between the test groups and the metrics group is illustrated in Figure 8.

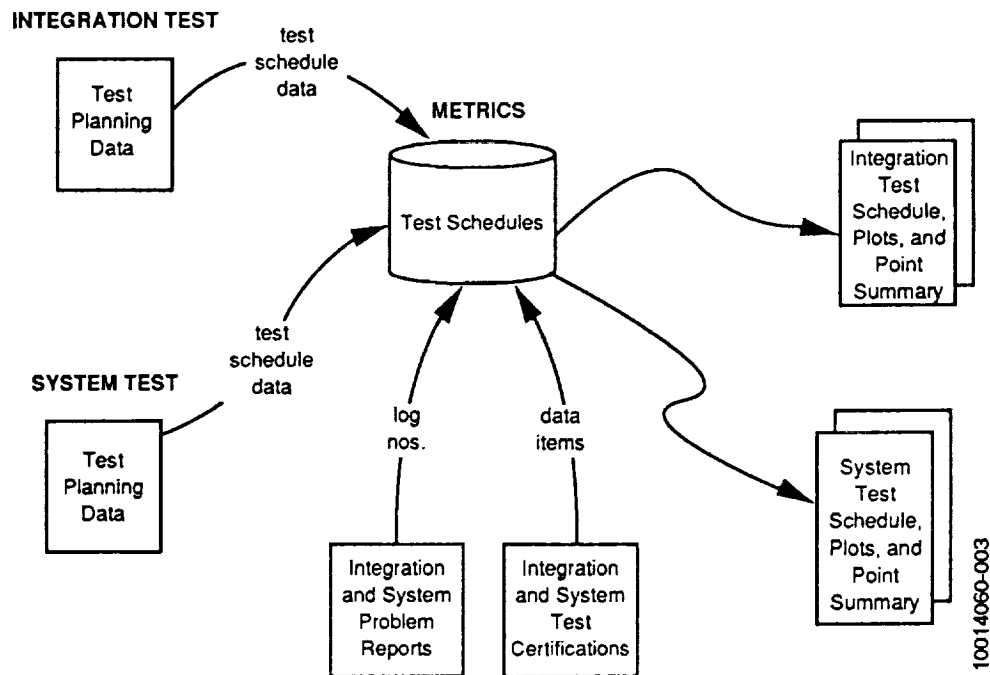


Figure 8. Data Flow - Test and Metrics

As previously stated, at the start of testing the baseline is put under CM control. During integration testing all problems found in the build or release are documented on Integration Software Problem Reports (ISPRs). During system testing, all problems are documented on Software Problem Reports (SPRs). The differentiation distinguishes the test phase in which a problem was found in a build or release.

Similar to the development and maintenance implementation activities, three basic products are provided in support of integration testing and system testing: a schedule of test activities, plots of the progress, and points summary data. Before the start of integration and system test, the integration or system test group provides information on the test cases that will be run against the build or release. Both test groups provide an itemized list with titles and numbers of each test to be run, along with the corresponding CSCI or release requirement. Also provided is additional information including scheduled dates for the start and/or completion of each test, the staff members responsible for each test, and the number of points to be allocated to each test. The use of these tools supports testers and test managers by facilitating the planning of resource allocation for specific tests, identifying problem reports

that are impacting test case progress, and providing weekly feedback on progress. The information collected also helps identify where test procedures are lacking the depth needed to thoroughly test the software.

BENEFITS OF THE NCC METRICS DATA COLLECTION ACTIVITY

NCC metrics reporting makes project status accessible, traceable, and concise. The metrics processes and tools are simple, yet flexible enough to accommodate the specific needs of different managers; the outputs can easily be tailored to each group's needs. Additionally, the use of a project-wide metrics data collection and reporting activity provides an excellent source of information for defect causal analysis. Based on three years of practice, the benefits of the NCC metrics activity can be summarized into three major categories: *Planning*, *Monitoring and Control*, and *Defect Causal Analysis*.

Planning

From the management perspective, the initial and updated schedules provided by the metrics group identify work to be accomplished, in detail, before the start of the effort, for both implementation and testing. Managers are able to establish guidelines for the work to be accomplished during the scheduled interval of time. When used for planning, the implementation and test schedules indicate the concentration of items to be accomplished by date and by functional area. The distribution of staff resources can be mapped and then adjusted as necessary. The use of detailed schedules facilitates the formulation of a workable and realistic plan. From the technical perspective, once the schedules are established they are made available for reference. The plan of action is clear not only to management, but also to those directly responsible for accomplishing the work. Individuals can formulate their own plan for accomplishing work for which they are responsible. Making the schedules available to technical staff also facilitates communication. Each person knows who is responsible for specific items, therefore questions and information can be directed appropriately.

Monitoring and Control

Each manager involved in the completion of an NCC software build or release is required to plan his or her work. Therefore, it is also incumbent upon the managers to compare their planned activities to the progress being made. The points summary and the plots provide at-a-glance feedback on planned versus actual progress. This assists managers in

preparing for monthly status reviews. Metrics reports provide access to historical data that is used as a basis for planning future software implementations.

The regular distribution of metrics reports allows managers and technical staff to identify potential problems as they are developing. It is possible to apply a mitigation action before a problem grows in magnitude. The continuous data capture and reporting cycles facilitate the monitoring and control effort and direct managers to specific areas of concern. Metrics processes track the progress of the implementation down to the unit level. Items that are significantly behind schedule are flagged for further investigation. Similarly, during problem resolution, the progress of test cases and of problem resolution is closely tracked through the data collection and reporting process. The progress of all activities - development, testing and problem resolution - are tied to points summaries and plots. Therefore there are several levels at which information is reported. Plots illustrate progress, and points summaries numerically represent the progress and provide the basis for taking credit for accomplishments on a monthly basis. The schedules contain the detailed information needed by line managers and task leaders.

Defect Causal Analysis

An important initiative in the SEAS program is the defect causal analysis of software implementation and testing efforts. The NCC Project developed a DCA procedure based on data collected by the metrics task, and additional analysis provided by the technical staff. The metrics group provides key data collection and reporting DCA on the NCC project.

In addition to the three basic products already described, the initiation and resolution of ISPRs and SPRs is monitored using the *Detailed Defect Causal Analysis (DCA) Listing*. This spreadsheet and associated plots contain information such as when a problem report was written, the affected NCCDS segment, the manager or task leader assigned to analyze and resolve the problem, and data items that characterize the problem resolution. To aid in DCA, plots are generated that illustrate the characteristics of the problem report resolutions applied to a build or release. When performing DCA of each build or release, it is often helpful to make comparisons of the results against previous build or release statistics. Metrics reports draw attention to areas that are consistently at risk in each subsequent build or release. Results are fed into the subsequent planning process in order to formulate risk mitigation approaches.

An analysis of Release 92.1 statistics by the metrics and development groups for the Service Planning Segment (SPS) of the NCCDS showed that for software items against which formal reviews were conducted by development during the design and code of the release, no software changes were necessary in integration or system testing. As a result of these findings, the SPS group has enhanced its internal review procedures.

PROCESS IMPROVEMENT YIELDS QUALITY IMPROVEMENT

The effect of the project metrics activity on the quality of the NCCDS software is best illustrated in the following chart , Figure 9, that compares size and incidence of errors for four recent NCCDS Releases. Release 90.1 and Release 91.1, software maintenance releases, were implemented and tested prior to the start of the NCC Metrics activity. The metrics activity was initiated with the first build of Release 92.1, a two build software development release. The first maintenance activity to be included in the NCCDS system of metrics was Release 93.1.

On this chart, SPRs are software problem reports written by the NCCDS Project before delivery to the customer, STRs are system trouble reports written by the GFSC acceptance test team. The statistics show that since the advent of the NCC metrics system

development Release 92.1, with the largest number of delivered source instructions, has the lowest overall error rate, and

maintenance Release 93.1, at almost twice the size of Release 91.1, was delivered with half as many total SPRs and STRs.

Release Identifier	90.1 (Maint) Before Metrics	91.1 (Maint) Before Metrics	92.1 (Dev) After Metrics	93.1 (Maint) After Metrics
Release Size (DSI)	7925	18308	112115	36041
Total SPRs Reported	31	135	221	155
Total STRs Reported	15	34	92	25
SPR Errors/KDSI	3.91	7.37	1.97	4.30
STR Errors/KDSI	1.89	1.85	0.82	0.69

Figure 9. Release Size and Error Rate Comparison

Since the system of metrics data collection and reporting has been in use, the rate of errors per thousand DSI has decreased. Collecting defect-related statistics on the schedule update report, the testing schedules and plots, and on the detailed DCA listing has helped to focus attention on critical areas of the software baseline, this has aided in the resolution of problems and the unmasking of additional problems before delivery to the customer. Also, by using the schedules, plots and points summaries to navigate development and test efforts, the NCC Project has met the majority of its internal milestones, and made all of its scheduled deliveries to the customer on time.

SUMMARY

The goal of the NCC Project metrics activity has been to support the project processes and procedures in order that each build or release be delivered on schedule and reflective of high quality. Consistent with this goal, the objectives to be met are to establish plans, monitor the progress according to the plan, and utilize the feedback to effectively manage progress, growth and change during the implementation and test phases. In addition to the above benefits of the metrics data collection and reporting processes, data have been used in the development of Baseline History Reports, and as evidence in internal, division level, and GSFC process audits. Models of the NCC Metrics plan have been used in contract proposals to outline a method for supporting the software development lifecycle.

Comparisons of NCCDS release histories, and the increased level of customer satisfaction have proven that the use of simple tools to support management and technical staff, as described in this paper, have had a measurable effect on the ability of the NCC project to deliver high quality, error-free builds and releases.

REFERENCES

1. Computer Sciences Corporation, *SEAS Software Development Methodology (SSDM)*
2. _____, *Network Control Center Data System (NCCDS) Metrics Handbook*, Draft, September 1993
3. _____, *NCC Standards and Procedures*, Revision 7, March 1993
4. _____, *Release 90.1 Baseline History Report*, February 1991
5. _____, *Release 91.1 Baseline History Report*, January, 1992
6. _____, *Release 92.1 Baseline History Report*, February 1993

